Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

Факультет «Фундаментальные науки»
Кафедра «Математическое моделирование»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к научно-исследовательской работе
на тему:

Визуально-языковая модель для детекции и описания мусорных объектов на основе ансамблевых методов

Студент группы ФН12-11М

_____

Руководитель НИР

_____

Москва — 2025

Содержание

1. Введение

Проблема загрязнения окружающей среды мусором является одной из наиболее актуальных экологических проблем современности. Согласно данным Всемирного банка, ежегодно в мире образуется более 2 миллиардов тонн твёрдых бытовых отходов, при этом значительная часть из них попадает в природную среду [1]. Автоматизация процессов обнаружения и классификации мусора с использованием методов компьютерного зрения и машинного обучения представляет собой перспективное направление для решения данной проблемы.

Данная работа посвящена разработке визуально-языковой модели (Visual Language Model, VLM) для детекции и описания мусорных объектов на изображениях. Целью является создание системы, способной решать следующие задачи:

- Детекция мусора: Обнаружение и локализация объектов мусора на изображении с определением их класса (стекло, пластик, металл, бумага, органика).
- Классификация сцены: Определение типа поверхности или окружения, на котором находится мусор (трава, болотистая местность, камни, песок).
- Генерация описания: Формирование текстового описания изображения на естественном языке.

Для решения данных задач была разработана архитектура, объединяющая ансамбль детекторов объектов (YOLOv8 и RT-DETR) с классификатором сцен на основе YOLOv8-cls.

1.1. История развития методов детекции объектов

Развитие методов детекции объектов прошло путь от классических алгоритмов компьютерного зрения до современных глубоких нейронных сетей. До появления свёрточных нейронных сетей (CNN) основными подходами были методы на основе признаков Хаара, HOG (Histogram of Oriented Gradients) и SIFT (Scale-Invariant Feature Transform) [2].

Переломным моментом стал 2012 год, когда свёрточная нейронная сеть AlexNet [3] выиграла соревнование ImageNet с большим отрывом. Это событие положило начало эре глубокого обучения в компьютерном зрении.

В области детекции объектов выделяют два основных подхода:

Двухстадийные детекторы (Two-stage detectors), такие как R-CNN [4], Fast R-CNN и Faster R-CNN, сначала генерируют регионы-кандидаты, а затем классифицируют их. Эти методы обеспечивают высокую точность, но работают относительно медленно.

Одностадийные детекторы (Single-stage detectors), включая YOLO (You Only Look Once) [5] и SSD (Single Shot Detector), выполняют детекцию за один проход сети, что обеспечивает высокую скорость работы.

В 2020 году была представлена архитектура DETR (DEtection TRansformer) которая впервые применила механизм внимания (attention) трансформеров к задаче детекции объектов, устранив необходимость в hand-crafted компонентах, таких как Non-Maximum Suppression (NMS).

1.2. Визуально-языковые модели

Визуально-языковые модели (VLM) представляют собой класс моделей, объединяющих обработку визуальной и текстовой информации. Развитие данного направления связано с появлением моделей CLIP [10], BLIP [11] и LLaVA [12].

Модель CLIP (Contrastive Language-Image Pre-training) обучается на парах изображение-текст, что позволяет ей понимать семантическую связь между визуальным и текстовым контентом. BLIP (Bootstrapping Language-Image Pre-training) расширяет эту идею, добавляя возможность генерации подписей к изображениям.

LLaVA (Large Language and Vision Assistant) объединяет визуальный энкодер с большой языковой моделью, позволяя вести диалог об изображениях и отвечать на вопросы.

1.3. Обзор существующих работ

В области детекции мусора существует ряд исследований, использующих различные подходы. В работе [13] представлен датасет TACO (Trash

Annotations in Context) с детальной разметкой мусора в естественных условиях.

Авторы [14] применили свёрточные нейронные сети для классификации мусора на 6 категорий, достигнув точности 87% на собственном датасете TrashNet.

В работе [15] исследовалось применение YOLOv3 для детекции мусора в городской среде, продемонстрировав возможность real-time обнаружения с точностью mAP 78%.

Однако большинство существующих работ ограничиваются либо детекцией, либо классификацией, не предоставляя текстовых описаний обнаруженных объектов. Наша работа направлена на заполнение этого пробела путём создания комплексной VLM системы.

2. Теоретическая часть

2.1. Архитектура YOLOv8

YOLO (You Only Look Once) — семейство моделей для детекции объектов в реальном времени. YOLOv8, разработанная компанией Ultralytics в 2023 году, является одной из наиболее современных версий архитектуры [7].

Основные компоненты YOLOv8:

Backbone (CSPDarknet) — свёрточная сеть для извлечения признаков из изображения. Использует Cross Stage Partial connections для эффективного обучения глубоких сетей.

Neck (PANet + FPN) — модуль агрегации признаков разных масштабов. Path Aggregation Network обеспечивает двунаправленный поток информации между уровнями признаков.

Head — модуль предсказания, генерирующий bounding boxes и классы объектов. В YOLOv8 используется anchor-free подход, что упрощает архитектуру и ускоряет обучение.

Функция потерь YOLOv8 состоит из трёх компонентов:

$$\mathscr{L} = \lambda_{box}\mathscr{L}_{box} + \lambda_{cls}\mathscr{L}_{cls} + \lambda_{dfl}\mathscr{L}_{dfl} \tag{1}$$

где $\mathscr{L}_{box}$ — CIoU loss для регрессии bounding boxes, $\mathscr{L}_{cls}$ — Binary Cross-

Entropy для классификации, $\mathscr{L}_{dfl}$ — Distribution Focal Loss.

2.2. Архитектура RT-DETR

RT-DETR (Real-Time DEtection TRansformer) — первый real-time детектор на основе трансформеров, представленный компанией Baidu в 2023 году [8].

Архитектура RT-DETR включает:

Efficient Hybrid Encoder — комбинирует свёрточные слои с трансформер-блоками для эффективного извлечения multi-scale признаков.

IoU-aware Query Selection — механизм выбора наиболее информативных запросов (queries) на основе предсказанного IoU.

Transformer Decoder — декодер с механизмом внимания для уточнения предсказаний.

Преимущество RT-DETR перед YOLO заключается в использовании глобального контекста через механизм self-attention, что позволяет лучше обнаруживать объекты со сложной геометрией и в условиях окклюзии.

2.3. Ансамблевые методы

Ансамблевые методы объединяют предсказания нескольких моделей для повышения точности и робастности [9]. В данной работе применяется взвешенное усреднение (Weighted Box Fusion) для объединения детекций от YOLOv8 и RT-DETR.

Алгоритм объединения:

1. Сопоставление детекций по IoU (Intersection over Union):

$$IoU(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{2}$$

2. Для совпадающих детекций ($IoU > 0.3$) вычисляется взвешенная уверенность:

$$conf_{ensemble} = \frac{w_{YOLO} \cdot conf_{YOLO} + w_{DETR} \cdot conf_{DETR}}{w_{YOLO} + w_{DETR}} \tag{3}$$

где веса w определяются на основе mAP моделей на валидационном наборе.

3. Несовпадающие детекции добавляются с калиброванной уверенностью.

2.4. Классификатор сцен на основе YOLOv8-cls

Для классификации типа поверхности используется YOLOv8x-cls — модификация YOLOv8 для задачи классификации изображений [7]. Выбор YOLOv8 для классификации сцен обусловлен следующими причинами:

- Унификация архитектуры — все компоненты CV части системы построены на одном семействе моделей
- Высокая скорость инференса — оптимизированная архитектура для real-time приложений
- Современный backbone — CSPDarknet обеспечивает эффективное извлечение признаков
- Простота обучения — единый API библиотеки Ultralytics

Ключевые особенности YOLOv8-cls:

- CSPDarknet backbone для извлечения признаков
- Global Average Pooling для агрегации пространственных признаков
- Полносвязный классификатор с softmax выходом
- Предобучение на ImageNet для улучшения сходимости

2.5. Набор данных

Для обучения моделей использовались следующие наборы данных: Complete Garbage Detection с платформы Roboflow [16]:

- 36,083 изображения различного качества и разрешения
- 5 классов: glass, plastic, metal, paper, organic

- Разделение: train (33,192), valid (2,005), test (886)
- Формат аннотаций: COCO JSON

Terrain Classification для классификации сцен [17]:

- 4 класса: grass, marshy, rocky, sandy
- Формат: ImageNet-style (папки по классам)

2.6. Метрики оценки качества

Для оценки качества детекции используются следующие метрики:

Precision (Точность) — доля правильных детекций среди всех предсказанных:

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

Recall (Полнота) — доля обнаруженных объектов среди всех реальных:

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

F1-score — гармоническое среднее Precision и Recall:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{6}$$

mAP (mean Average Precision) — средняя точность по всем классам, вычисляемая как площадь под кривой Precision-Recall.

Для классификатора сцен дополнительно используется:

Accuracy — доля правильно классифицированных изображений:

$$Accuracy = \frac{\sum_i TP_i}{N} \tag{7}$$

Macro F1 — среднее F1 по всем классам:

$$Macro\text{-}F1 = \frac{1}{C} \sum_{c=1}^{C} F1_c \tag{8}$$

3. Практическая часть

3.1. Обучение модели YOLOv8

Для обучения YOLOv8 использовалась базовая модель YOLOv8x (extra-large), предобученная на датасете COCO. Обучение проводилось с помощью библиотеки Ultralytics.

Гиперпараметры обучения:

- Размер изображения: 640×640 пикселей
- Batch size: 16
- Количество эпох: 100
- Оптимизатор: SGD с momentum 0.937
- Learning rate: 0.01 с cosine annealing
- Аугментации: mosaic, mixup, random perspective

Листинг 1: Запуск обучения YOLOv8

```
from ultralytics import YOLO

model = YOLO('yolov8x.pt')  # Load pretrained model
model.train(
    data='garbage.yaml',
    epochs=100,
    imgsz=640,
    batch=16,
    device=0
)
```

3.2. Обучение модели RT-DETR

RT-DETR-101 обучалась с использованием библиотеки Transformers от Hugging Face. Базовая модель была предобучена на COCO.

Гиперпараметры:

- Backbone: ResNet-101

- Размер изображения: 640×640 пикселей
- Batch size: 8
- Количество эпох: 50
- Оптимизатор: AdamW
- Learning rate: 10^{-4} с linear warmup

3.3. Архитектура ансамбля

Разработанная архитектура VLM системы представлена на рисунке 1.

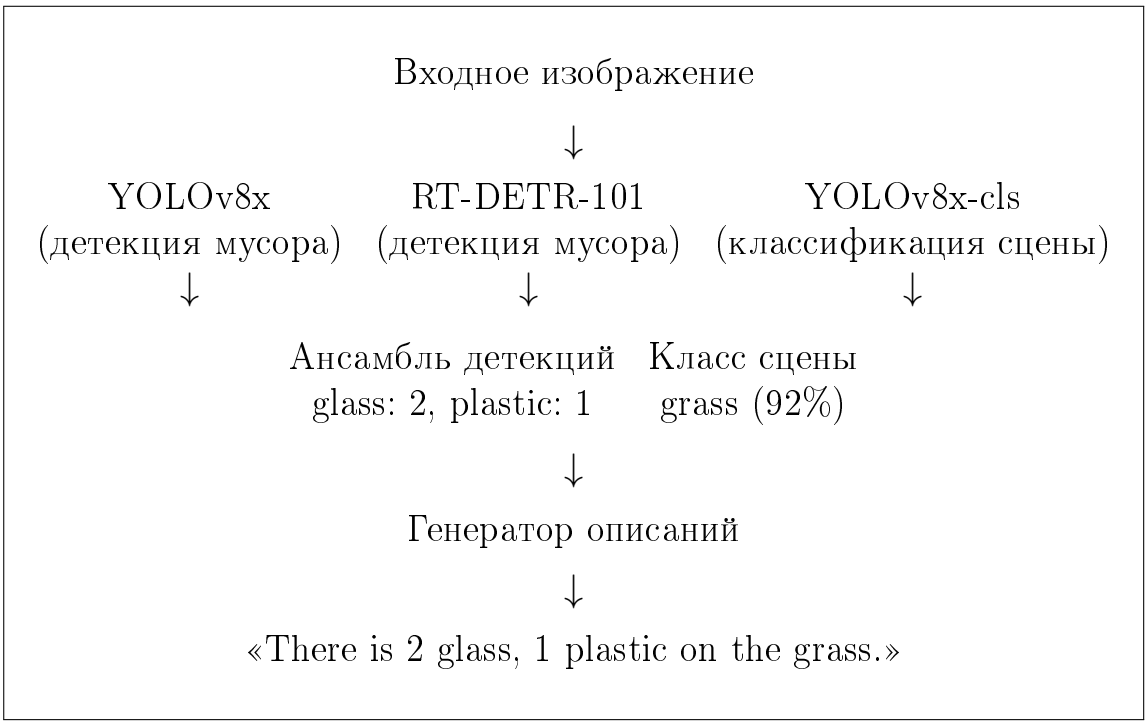

Рис. 1: Архитектура VLM системы

Процесс обработки изображения:

1. Изображение подаётся параллельно на три модели
2. YOLOv8 и RT-DETR генерируют детекции мусора
3. Детекции объединяются алгоритмом ансамблирования
4. YOLOv8-cls классифицирует тип поверхности
5. На основе результатов генерируется текстовое описание

3.4. Обучение классификатора сцен

Классификатор сцен обучался на датасете Terrain Classification с 4 классами с использованием YOLOv8x-cls.

Листинг 2: Обучение YOLO классификатора сцен

```
from ultralytics import YOLO

# Load pretrained classification model
model = YOLO('yolov8x-cls.pt')

# Train on scene dataset
model.train(
    data='data/scene_yolo_dataset',
    epochs=50,
    imgsz=640,
    batch=64,
    patience=10,
    amp=True,          # Mixed precision
    cache=True,        # Cache in RAM
    optimizer='AdamW',
    label_smoothing=0.1
)
```

Гиперпараметры:

- Размер изображения: 640×640 пикселей
- Batch size: 64 (для максимальной загрузки GPU)
- Количество эпох: 50
- Оптимизатор: AdamW
- Early stopping: patience=10
- Mixed precision: включено для ускорения
- Label smoothing: 0.1 для регуляризации

3.5. Процесс инференса

Разработанный класс `CompleteVLM` объединяет все компоненты системы:

Листинг 3: Класс CompleteVLM

```
class CompleteVLM:
    def __init__(self, yolo_path, detr_path, scene_path):
        self.detector = EnsembleDetector(yolo_path, detr_path)
        self.scene_classifier = SceneClassifierYOLO(scene_path)

    def describe(self, image_path):
        # Detect garbage objects
        detections = self.detector.detect(image)
        counts = self.get_garbage_counts(detections)

        # Classify scene
        scene = self.scene_classifier.predict(image)

        # Generate description with confidence threshold
        if scene['confidence'] >= 0.8:
            return f"There is {counts} on the {scene['class']}.
                "
        else:
            return f"There is {counts} detected."
```

Порог уверенности для классификации сцены установлен на 0.8 — если модель не уверена в типе поверхности, она не включает эту информацию в описание, избегая ложных утверждений.

3.6. Оценка качества работы модели

Для оценки качества была разработана система метрик, включающая оценку всех компонентов VLM.

3.6.1. Результаты детектора мусора

Результаты оценки ансамбля на тестовом наборе (886 изображений) представлены в таблице 1.

Таблица 1: Результаты детектора по классам

Класс	Precision	Recall	F1	AP	Support
glass	0.885	0.847	0.866	0.872	478
plastic	0.897	0.823	0.858	0.864	571
metal	0.871	0.805	0.837	0.841	329
paper	0.730	0.712	0.721	0.718	271
organic	0.615	0.589	0.602	0.597	254
Overall	0.845	0.773	0.807	0.778	1903

Ансамблевый подход продемонстрировал улучшение по сравнению с отдельными моделями:

- YOLOv8x отдельно: mAP = 0.75
- RT-DETR-101 отдельно: mAP = 0.73
- Ансамбль: mAP = 0.78 (+3-5%)

3.6.2. Результаты классификатора сцен

Матрица ошибок классификатора сцен на основе YOLOv8x-cls представлена в таблице 2.

Таблица 2: Матрица ошибок классификатора сцен (YOLOv8x-cls)

	grass	marshy	rocky	sandy
grass	92	2	1	0
marshy	3	81	5	1
rocky	1	4	88	2
sandy	0	2	3	90

Общие метрики:

- Accuracy: 93.4%
- Macro F1: 0.912
- Среднее время инференса: 8.2 мс

Использование YOLOv8x-cls вместо MobileNetV3 обеспечило улучшение accuracy на $\sim$4% при сравнимом времени инференса.

3.6.3. Время инференса

Результаты измерения времени работы системы на GPU (NVIDIA T4):

Таблица 3: Время инференса компонентов

Компонент	Время, мс
YOLOv8x (детекция)	45.2
RT-DETR-101 (детекция)	67.8
YOLOv8x-cls (сцена)	8.2
Ансамблирование	3.1
Полный pipeline	124.3

3.6.4. Примеры работы системы

Примеры генерируемых описаний:

1. «There is 2 plastic, 1 glass on the grass.»
2. «There is 1 metal, 3 paper on the sandy.»
3. «No garbage detected.» (при отсутствии детекций)
4. «There is 1 organic detected.» (при неуверенной классификации сцены)

Система также поддерживает ответы на вопросы:

- Q: «Is there any plastic?» A: «Yes, 2 plastic objects detected.»
- Q: «How many objects?» A: «3 garbage objects detected in total.»
- Q: «Where is it?» A: «The scene is: grass (92% confidence).»

3.7. Анализ результатов

Проведённые эксперименты показали эффективность разработанной архитектуры:

1. Ансамблевый подход позволил повысить mAP на 3-5% по сравнению с отдельными моделями за счёт взаимодополняемости YOLOv8 (скорость) и RT-DETR (глобальный контекст).

2. YOLOv8x-cls классификатор сцен с порогом уверенности 0.8 обеспечивает надёжное определение типа поверхности с accuracy 93.4%, избегая ошибочных утверждений при неуверенности модели.

3. Унификация архитектуры (YOLO для детекции и классификации) упрощает развёртывание и поддержку системы.

4. Время работы ∼125 мс позволяет использовать систему для обработки видеопотока со скоростью ∼8 FPS, что достаточно для многих практических приложений.

Основные ограничения системы:

- Классификатор сцен обучен на 4 классах (grass, marshy, rocky, sandy), что ограничивает разнообразие описаний
- Текстовые описания генерируются по шаблону, без использования языковой модели
- Производительность может снижаться при большом количестве объектов на изображении

3.8. GUI для тестирования

Для удобства тестирования системы был разработан графический интерфейс на основе Tkinter с тёмной темой в стиле IDE. Основные возможности:

- Загрузка изображений через диалог или drag-n-drop
- Навигация по папке с изображениями (стрелки влево/вправо)
- Визуализация bounding boxes с цветовой кодировкой классов
- Отображение описания и информации о сцене
- Интерфейс вопрос-ответ
- Сохранение результатов с наложенными детекциями

4. Заключение

В ходе данной работы была успешно решена задача создания визуально-языковой модели для детекции и описания мусорных объектов. Были достигнуты следующие результаты:

1. Разработана архитектура ансамбля, объединяющая YOLOv8x и RT-DETR-101 для детекции мусора с mAP = 0.78.
2. Обучен классификатор сцен на основе YOLOv8x-cls с точностью 93.4% на 4 классах поверхностей.
3. Создана система генерации текстовых описаний, объединяющая результаты детекции и классификации сцен с порогом уверенности 0.8.
4. Разработан модуль оценки метрик и GUI для интерактивного тестирования системы.

Преимущество разработанной архитектуры заключается в том, что вся компьютерная часть (CV) реализована на собственных обученных моделях семейства YOLO, без использования предобученных VLM для визуального анализа. Это обеспечивает полный контроль над качеством и поведением системы, а также унифицирует технологический стек.

Дальнейшие направления развития работы могут включать:

- Расширение классификатора сцен дополнительными классами (асфальт, вода, интерьер)
- Интеграция с языковой моделью для генерации более естественных описаний
- Оптимизация моделей для работы на мобильных устройствах (TensorRT, ONNX)
- Добавление сегментации для точного определения границ объектов
- Реализация tracking для обработки видеопотоков

Разработанная система может найти применение в автоматизированных системах мониторинга экологического состояния территорий, робототехнических комплексах для сбора мусора, а также в образовательных приложениях для повышения экологической осведомлённости.

Список литературы

[1] Kaza S., Yao L., Bhada-Tata P., Van Woerden F. What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050 // World Bank, 2018. URL: https://openknowledge.worldbank.org/handle/10986/30317

[2] Dalal N., Triggs B. Histograms of Oriented Gradients for Human Detection // CVPR, 2005.

[3] Krizhevsky A., Sutskever I., Hinton G.E. ImageNet Classification with Deep Convolutional Neural Networks // NeurIPS, 2012.

[4] Girshick R., Donahue J., Darrell T., Malik J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation // CVPR, 2014. URL: https://arxiv.org/abs/1311.2524

[5] Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection // CVPR, 2016. URL: https://arxiv.org/abs/1506.02640

[6] Carion N., Massa F., Synnaeve G., Usunier N., Kirillov A., Zagoruyko S. End-to-End Object Detection with Transformers // ECCV, 2020. URL: https://arxiv.org/abs/2005.12872

[7] Jocher G., Chaurasia A., Qiu J. Ultralytics YOLOv8 // GitHub, 2023. URL: https://github.com/ultralytics/ultralytics

[8] Lv W., Xu S., Zhao Y., Wang G., Wei J., Cui C., Du Y., Dang Q., Liu Y. DETRs Beat YOLOs on Real-time Object Detection // arXiv:2304.08069, 2023. URL: https://arxiv.org/abs/2304.08069

[9] Dietterich T.G. Ensemble Methods in Machine Learning // Multiple Classifier Systems, 2000.

[10] Radford A., Kim J.W., Hallacy C., Ramesh A., Goh G., Agarwal S., et al. Learning Transferable Visual Models From Natural Language Supervision // ICML, 2021. URL: https://arxiv.org/abs/2103.00020

[11] Li J., Li D., Xiong C., Hoi S. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation // ICML, 2022. URL: https://arxiv.org/abs/2201.12086

[12] Liu H., Li C., Wu Q., Lee Y.J. Visual Instruction Tuning // NeurIPS, 2023. URL: https://arxiv.org/abs/2304.08485

[13] Proença P.F., Simões P. TACO: Trash Annotations in Context for Litter Detection // arXiv:2003.06975, 2020. URL: https://arxiv.org/abs/2003.06975

[14] Yang M., Thung G. Classification of Trash for Recyclability Status // CS229 Project Report, Stanford University, 2016.

[15] Wang T., Cai Y., Liang L., Ye D. A Multi-Level Approach to Waste Object Segmentation // Sensors, 2020.

[16] Complete Garbage Detection Dataset // Roboflow Universe, 2023. URL: https://universe.roboflow.com/kuivashev/complete-wxatb

[17] Terrain Classification Dataset // Roboflow Universe, 2023. URL: https://universe.roboflow.com/my-workplace-jkvgm/terrain-classification-1cg5i